Probabilistic & Unsupervised Learning
Approximate Inference

Bayesian Model selection, Hyperparameter optimisation, and Gaussian Processes

Maneesh Sahani

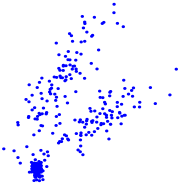
maneesh@gatsby.ucl.ac.uk

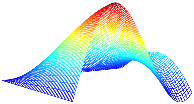
Gatsby Computational Neuroscience Unit, and
MSc ML/CSML, Dept Computer Science
University College London

Term 1, Autumn 2020

Learning model structure

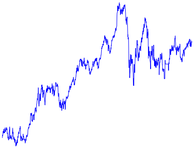
How many clusters in the data?

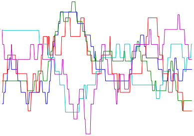




How smooth should the function be?

Is this input relevant to predicting that output?



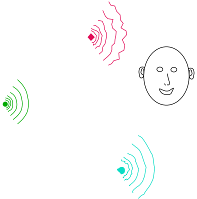


What is the order of a dynamical system?

How many states in a hidden Markov model?

SVYDAAAQLTADVKKDLRDSWKVIGSDKKGNG

How many auditory sources in the input?



Model selection

Models (labelled by m) have parameters $\boldsymbol{\theta}_m$ that specify the probability of data:

$$P(\mathcal{D}|\boldsymbol{\theta}_m, m)\,.$$

If model is known, learning $\boldsymbol{\theta}_m$ means finding posterior or point estimate (ML, MAP, ...).

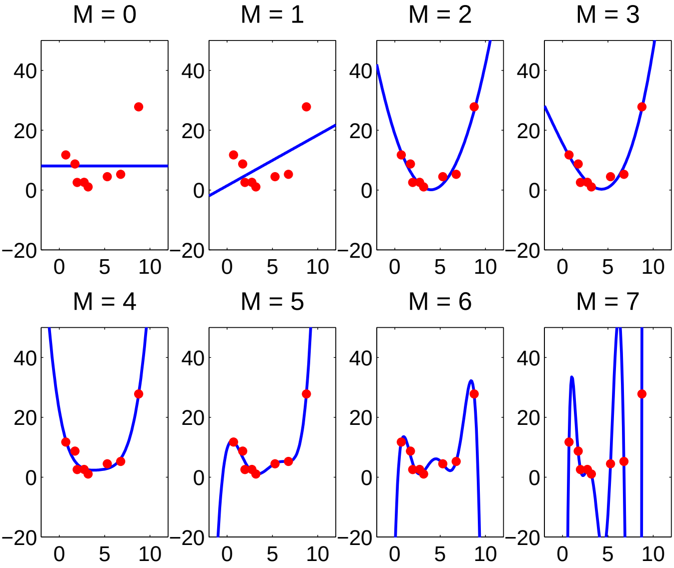
What if we need to learn the model too?

- Could combine models into a single "supermodel", with composite parameter $(m, \boldsymbol{\theta}_m)$.
  - ML learning will overfit: favours most flexible (nested) model with most parameters, even if the data actually come from a simpler one.
  - Density function on composite parameter space (union of manifolds of different dimensionalities) difficult to define $\Rightarrow$ MAP learning ill-posed.
  - Joint posterior difficult to compute — dimension of composite parameter varies [although Monte-Carlo methods may be able to sample from such a posterior.]

$\Rightarrow$ Separate model selection step:

$$P(\boldsymbol{\theta}_m, m|\mathcal{D}) = \underbrace{P(\boldsymbol{\theta}_m|m, \mathcal{D})}_{\text{model-specific posterior}} \cdot \underbrace{P(m|\mathcal{D})}_{\text{model selection}}$$

Model complexity and overfitting: a simple example



Model selection

Given models labeled by m with parameters $\boldsymbol{\theta}_m$, identify the "correct" model for data $\mathcal{D}$.

ML/MAP has no good answer: $P(\mathcal{D}|\boldsymbol{\theta}_m^{\mathsf{ML}})$ is always larger for more complex (nested) models.

Model selection

Given models labeled by m with parameters $\boldsymbol{\theta}_m$, identify the “correct” model for data $\mathcal{D}$.

ML/MAP has no good answer: $P(\mathcal{D}|\boldsymbol{\theta}_m^{\text{ML}})$ is always larger for more complex (nested) models.

Neyman-Pearson hypothesis testing

- For nested models. Starting with simplest model ($m = 1$), compare (e.g. by likelihood ratio test) null hypothesis m to alternative $m + 1$. Continue until $m + 1$ is rejected.
- Tests often only exact asympotically in data number.
- Conservative (N-P hypothesis tests are asymmetric by design).

Model selection

Given models labeled by m with parameters $\boldsymbol{\theta}_m$, identify the "correct" model for data $\mathcal{D}$.

ML/MAP has no good answer: $P(\mathcal{D}|\boldsymbol{\theta}_m^{\mathrm{ML}})$ is always larger for more complex (nested) models.

Neyman-Pearson hypothesis testing

- For nested models. Starting with simplest model ($m = 1$), compare (e.g. by likelihood ratio test) null hypothesis m to alternative $m + 1$. Continue until $m + 1$ is rejected.
- Tests often only exact asympotically in data number.
- Conservative (N-P hypothesis tests are asymmetric by design).

Likelihood validation

- Partition data into disjoint *training* and *validation* data sets $\mathcal{D} = \mathcal{D}_{\mathrm{tr}} \cup \mathcal{D}_{\mathrm{vld}}$. Choose model with greatest $P(\mathcal{D}_{\mathrm{vld}}|\boldsymbol{\theta}_m^{\mathrm{ML}})$, with $\boldsymbol{\theta}_m^{\mathrm{ML}} = \operatorname{argmax} P(\mathcal{D}_{\mathrm{tr}}|\boldsymbol{\theta})$. [Or, better, greatest $P(\mathcal{D}_{\mathrm{vld}}|\mathcal{D}_{\mathrm{tr}}, m)$.]
- Consistent; and selects most useful model, even if all are incorrect.
- May be biased towards simpler models; often high-variance.
- Cross-validation uses multiple partitions and averages likelihoods.

Model selection

Given models labeled by m with parameters $\boldsymbol{\theta}_m$, identify the "correct" model for data $\mathcal{D}$.

ML/MAP has no good answer: $P(\mathcal{D}|\boldsymbol{\theta}_m^{\mathrm{ML}})$ is always larger for more complex (nested) models.

Neyman-Pearson hypothesis testing

- For nested models. Starting with simplest model ($m = 1$), compare (e.g. by likelihood ratio test) null hypothesis m to alternative $m + 1$. Continue until $m + 1$ is rejected.
- Tests often only exact asympotically in data number.
- Conservative (N-P hypothesis tests are asymmetric by design).

Likelihood validation

- Partition data into disjoint *training* and *validation* data sets $\mathcal{D} = \mathcal{D}_{\mathrm{tr}} \cup \mathcal{D}_{\mathrm{vld}}$. Choose model with greatest $P(\mathcal{D}_{\mathrm{vld}}|\boldsymbol{\theta}_m^{\mathrm{ML}})$, with $\boldsymbol{\theta}_m^{\mathrm{ML}} = \operatorname{argmax} P(\mathcal{D}_{\mathrm{tr}}|\boldsymbol{\theta})$. [Or, better, greatest $P(\mathcal{D}_{\mathrm{vld}}|\mathcal{D}_{\mathrm{tr}}, m)$.]
- Consistent; and selects most useful model, even if all are incorrect.
- May be biased towards simpler models; often high-variance.
- Cross-validation uses multiple partitions and averages likelihoods.

Bayesian model selection

- Choose most likely model: $\operatorname{argmax} P(m|\mathcal{D})$.
- Consistent; probabilistically principled if true model is in set being considered, but sensitive to assumed priors etc.
- Posterior probabilities can weight models for combined predictions (avoiding selection).

Bayesian model selection: some terminology

A model class m is a set of distributions parameterised by $\boldsymbol{\theta}_m$, e.g. the set of all possible mixtures of m Gaussians.

The model implies both a prior over the parameters $P(\boldsymbol{\theta}_m|m)$, and a likelihood of data given parameters (which might require integrating out latent variables) $P(\mathcal{D}|\boldsymbol{\theta}_m, m)$.

The posterior distribution over parameters is

$$P(\boldsymbol{\theta}_m|\mathcal{D}, m) = \frac{P(\mathcal{D}|\boldsymbol{\theta}_m, m)P(\boldsymbol{\theta}_m|m)}{P(\mathcal{D}|m)}.$$

The marginal probability of the data under model class m is:

$$P(\mathcal{D}|m) = \int_{\Theta_m} P(\mathcal{D}|\boldsymbol{\theta}_m, m)P(\boldsymbol{\theta}_m|m)\, d\boldsymbol{\theta}_m.$$

(also called the Bayesian evidence for model m).

The ratio of two marginal probabilities (or sometimes its log) is known as the Bayes factor:

$$\frac{P(\mathcal{D}|m)}{P(\mathcal{D}|m')} = \frac{P(m|\mathcal{D})}{P(m'|\mathcal{D})}\frac{p(m')}{p(m)}$$

The Bayesian Occam's razor

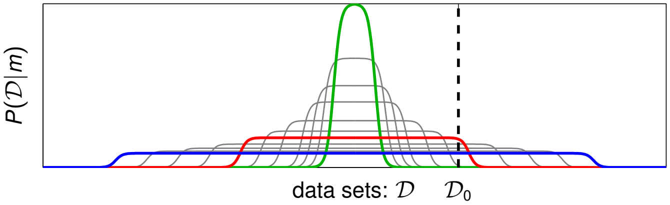
Occam's Razor is a principle of scientific philosophy: of two explanations adequate to explain the same set of observations, the simpler should always be preferred.
Bayesian inference formalises and *automatically* implements a form of Occam's Razor.

Compare model classes m using their posterior probability given the data:

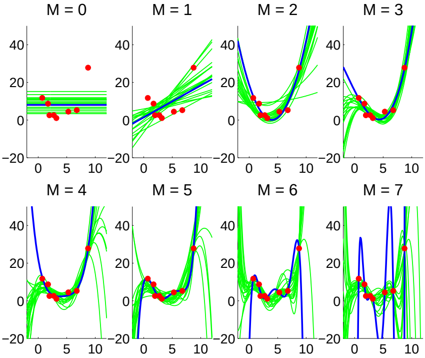
$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}, \qquad P(\mathcal{D}|m) = \int_{\Theta_m} P(\mathcal{D}|\boldsymbol{\theta}_m, m)P(\boldsymbol{\theta}_m|m)\, d\boldsymbol{\theta}_m$$

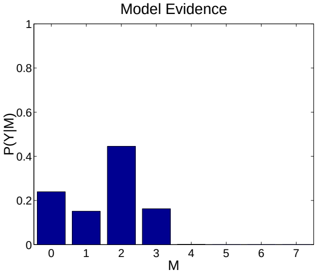
$P(\mathcal{D}|m)$: The probability that *randomly selected* parameter values from the model class would generate data set $\mathcal{D}$.

Model classes that are too simple are unlikely to generate the observed data set.
Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.
Like Goldilocks, we favour a model that is just right.



Bayesian model comparison: Occam's razor at work





Conjugate-exponential families (recap)

Can we compute $P(\mathcal{D}|m)$?

Conjugate-exponential families (recap)

Can we compute $P(\mathcal{D}|m)$? . . . . . . Sometimes.

Conjugate-exponential families (recap)

Can we compute $P(\mathcal{D}|m)$? ...... Sometimes.

Suppose $P(\mathcal{D}|\boldsymbol{\theta}_m, m)$ is a member of the exponential family:

$$P(\mathcal{D}|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} e^{\mathbf{s}(\mathbf{x}_i)^\top \boldsymbol{\theta}_m - A(\boldsymbol{\theta}_m)}.$$

Conjugate-exponential families (recap)

Can we compute $P(\mathcal{D}|m)$? ...... Sometimes.

Suppose $P(\mathcal{D}|\boldsymbol{\theta}_m, m)$ is a member of the exponential family:

$$P(\mathcal{D}|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} e^{\mathbf{s}(\mathbf{x}_i)^{\mathsf{T}}\boldsymbol{\theta}_m - A(\boldsymbol{\theta}_m)}.$$

If our prior on $\boldsymbol{\theta}_m$ is conjugate:

$$P(\boldsymbol{\theta}_m|m) = e^{\mathbf{s}_p^{\mathsf{T}}\boldsymbol{\theta}_m - n_p A(\boldsymbol{\theta}_m)} / Z(\mathbf{s}_p, n_p)$$

Conjugate-exponential families (recap)

Can we compute $P(\mathcal{D}|m)$? ...... Sometimes.

Suppose $P(\mathcal{D}|\boldsymbol{\theta}_m, m)$ is a member of the exponential family:

$$P(\mathcal{D}|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} e^{\mathbf{s}(\mathbf{x}_i)^\top \boldsymbol{\theta}_m - A(\boldsymbol{\theta}_m)}.$$

If our prior on $\boldsymbol{\theta}_m$ is conjugate:

$$P(\boldsymbol{\theta}_m|m) = e^{\mathbf{s}_p^\top \boldsymbol{\theta}_m - n_p A(\boldsymbol{\theta}_m)} / Z(\mathbf{s}_p, n_p)$$

then the joint is in the same family:

$$P(\mathcal{D}, \boldsymbol{\theta}_m|m) = e^{\left(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p\right)^\top \boldsymbol{\theta}_m - (N+n_p)A(\boldsymbol{\theta}_m)} / Z(\mathbf{s}_p, p)$$

Conjugate-exponential families (recap)

Can we compute $P(\mathcal{D}|m)$? . . . . . . Sometimes.

Suppose $P(\mathcal{D}|\boldsymbol{\theta}_m, m)$ is a member of the exponential family:

$$P(\mathcal{D}|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} e^{\mathbf{s}(\mathbf{x}_i)^\top \boldsymbol{\theta}_m - A(\boldsymbol{\theta}_m)}.$$

If our prior on $\boldsymbol{\theta}_m$ is conjugate:

$$P(\boldsymbol{\theta}_m|m) = e^{\mathbf{s}_p^\top \boldsymbol{\theta}_m - n_p A(\boldsymbol{\theta}_m)} / Z(\mathbf{s}_p, n_p)$$

then the joint is in the same family:

$$P(\mathcal{D}, \boldsymbol{\theta}_m|m) = e^{\left(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p\right)^\top \boldsymbol{\theta}_m - (N+n_p)A(\boldsymbol{\theta}_m)} / Z(\mathbf{s}_p, p)$$

and so:

$$P(\mathcal{D}|m) = \int d\boldsymbol{\theta}_m \; P(\mathcal{D}, \boldsymbol{\theta}_m|m) = \frac{Z\left(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p, N + n_p\right)}{Z(\mathbf{s}_p, p)}$$

Conjugate-exponential families (recap)

Can we compute $P(\mathcal{D}|m)$? ......Sometimes.

Suppose $P(\mathcal{D}|\boldsymbol{\theta}_m, m)$ is a member of the exponential family:

$$P(\mathcal{D}|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} e^{\mathbf{s}(\mathbf{x}_i)^{\top}\boldsymbol{\theta}_m - A(\boldsymbol{\theta}_m)}.$$

If our prior on $\boldsymbol{\theta}_m$ is conjugate:

$$P(\boldsymbol{\theta}_m|m) = e^{\mathbf{s}_p^{\top}\boldsymbol{\theta}_m - n_p A(\boldsymbol{\theta}_m)}/Z(\mathbf{s}_p, n_p)$$

then the joint is in the same family:

$$P(\mathcal{D}, \boldsymbol{\theta}_m|m) = e^{\left(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p\right)^{\top}\boldsymbol{\theta}_m - (N+n_p)A(\boldsymbol{\theta}_m)}/Z(\mathbf{s}_p, p)$$

and so:

$$P(\mathcal{D}|m) = \int d\boldsymbol{\theta}_m \, P(\mathcal{D}, \boldsymbol{\theta}_m|m) = \frac{Z\left(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p, N + n_p\right)}{Z(\mathbf{s}_p, p)} \begin{array}{ll} \leftarrow & \text{posterior volume} \\ \leftarrow & \text{prior volume} \end{array}$$

Conjugate-exponential families (recap)

Can we compute $P(\mathcal{D}|m)$? . . . . . . Sometimes.

Suppose $P(\mathcal{D}|\boldsymbol{\theta}_m, m)$ is a member of the exponential family:

$$P(\mathcal{D}|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} e^{\mathbf{s}(\mathbf{x}_i)^\top \boldsymbol{\theta}_m - A(\boldsymbol{\theta}_m)}.$$

If our prior on $\boldsymbol{\theta}_m$ is conjugate:

$$P(\boldsymbol{\theta}_m|m) = e^{\mathbf{s}_p^\top \boldsymbol{\theta}_m - n_p A(\boldsymbol{\theta}_m)} / Z(\mathbf{s}_p, n_p)$$

then the joint is in the same family:

$$P(\mathcal{D}, \boldsymbol{\theta}_m|m) = e^{\left(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p\right)^\top \boldsymbol{\theta}_m - (N+n_p)A(\boldsymbol{\theta}_m)} / Z(\mathbf{s}_p, p)$$

and so:

$$P(\mathcal{D}|m) = \int d\boldsymbol{\theta}_m \; P(\mathcal{D}, \boldsymbol{\theta}_m|m) = \frac{Z\left(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p, N + n_p\right)}{Z(\mathbf{s}_p, p)} \quad \begin{array}{ll} \leftarrow & \text{posterior volume} \\ \leftarrow & \text{prior volume} \end{array}$$

While the intuition is general, tractability is a special case. Thus, we must approximate . . .

Practical Bayesian approaches

- Laplace approximation:
  - Approximate posterior by a Gaussian centred at the maximum *a posteriori* parameter estimate.
- Bayesian Information Criterion (BIC)
  - an asymptotic ($N \to \infty$) approximation.
- Variational Bayes
  - Lower bound on the marginal probability.
  - Biased estimate.
  - Easy and fast, and often better than Laplace or BIC.
- Monte Carlo methods:
  - (Annealed) Importance sampling: estimate evidence using samples $\boldsymbol{\theta}^{(i)}$ from arbitrary $f(\boldsymbol{\theta})$:

$$\sum_i \frac{P(\mathcal{D}|\boldsymbol{\theta}^{(i)}, m)P(\boldsymbol{\theta}^{(i)}|m)}{f(\boldsymbol{\theta}^{(i)})} \to \int d\boldsymbol{\theta}\, f(\boldsymbol{\theta}) \frac{P(\mathcal{D}, \boldsymbol{\theta}|m)}{f(\boldsymbol{\theta})} = P(\mathcal{D}|m)$$

  - "Reversible jump" Markov Chain Monte Carlo: sample from posterior on composite $(m, \boldsymbol{\theta}_m)$. # samples for each $m \propto p(m|\mathcal{D})$.
  - Both exact in the limit of infinite samples, but may have high variance with finite samples.
- Bethe approximations, Expectation propagation ...

We will encounter many of these approaches later in the course.

Laplace approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\ d\boldsymbol{\theta}_m$.

Laplace approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\ d\boldsymbol{\theta}_m$.

As data size N grows (relative to parameter count d), $\boldsymbol{\theta}_m$ becomes more constrained
$\Rightarrow P(\mathcal{D}, \boldsymbol{\theta}_m|m) \propto P(\boldsymbol{\theta}_m|\mathcal{D}, m)$ becomes concentrated on posterior mode $\boldsymbol{\theta}_m^*$.

Laplace approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\; d\boldsymbol{\theta}_m$.

As data size N grows (relative to parameter count d), $\boldsymbol{\theta}_m$ becomes more constrained
$\Rightarrow P(\mathcal{D}, \boldsymbol{\theta}_m|m) \propto P(\boldsymbol{\theta}_m|\mathcal{D}, m)$ becomes concentrated on posterior mode $\boldsymbol{\theta}_m^*$.

Idea: approximate log $P(\mathcal{D}, \boldsymbol{\theta}_m|m)$ to second-order around $\boldsymbol{\theta}^*$.

Laplace approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\ d\boldsymbol{\theta}_m$.

As data size N grows (relative to parameter count d), $\boldsymbol{\theta}_m$ becomes more constrained
$\Rightarrow P(\mathcal{D}, \boldsymbol{\theta}_m|m) \propto P(\boldsymbol{\theta}_m|\mathcal{D}, m)$ becomes concentrated on posterior mode $\boldsymbol{\theta}_m^*$.

Idea: approximate log $P(\mathcal{D}, \boldsymbol{\theta}_m|m)$ to second-order around $\boldsymbol{\theta}^*$.

$$\int P(\mathcal{D}, \boldsymbol{\theta}_m|m) d\boldsymbol{\theta}_m = \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m|m)}\ d\boldsymbol{\theta}_m$$

Laplace approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\; d\boldsymbol{\theta}_m$.

As data size N grows (relative to parameter count d), $\boldsymbol{\theta}_m$ becomes more constrained
$\Rightarrow P(\mathcal{D}, \boldsymbol{\theta}_m|m) \propto P(\boldsymbol{\theta}_m|\mathcal{D}, m)$ becomes concentrated on posterior mode $\boldsymbol{\theta}^*_m$.

Idea: approximate log $P(\mathcal{D}, \boldsymbol{\theta}_m|m)$ to second-order around $\boldsymbol{\theta}^*$.

$$\int P(\mathcal{D}, \boldsymbol{\theta}_m|m) d\boldsymbol{\theta}_m = \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m|m)}\; d\boldsymbol{\theta}_m$$

$$\approx \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}^*_m|m) + \nabla \log P(\mathcal{D}, \boldsymbol{\theta}^*_m|m)\cdot(\boldsymbol{\theta}_m - \boldsymbol{\theta}^*_m) + \frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}^*_m)^{\mathsf{T}} \nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}^*|m)(\boldsymbol{\theta}_m - \boldsymbol{\theta}^*_m)}\; d\boldsymbol{\theta}_m$$

Laplace approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\ d\boldsymbol{\theta}_m$.

As data size N grows (relative to parameter count d), $\boldsymbol{\theta}_m$ becomes more constrained
$\Rightarrow P(\mathcal{D}, \boldsymbol{\theta}_m|m) \propto P(\boldsymbol{\theta}_m|\mathcal{D}, m)$ becomes concentrated on posterior mode $\boldsymbol{\theta}^*_m$.

Idea: approximate log $P(\mathcal{D}, \boldsymbol{\theta}_m|m)$ to second-order around $\boldsymbol{\theta}^*$.

$$\int P(\mathcal{D}, \boldsymbol{\theta}_m|m) d\boldsymbol{\theta}_m = \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m|m)}\ d\boldsymbol{\theta}_m$$

$$\approx \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}^*_m|m) + \underbrace{\nabla \log P(\mathcal{D}, \boldsymbol{\theta}^*_m|m)}_{=0} \cdot (\boldsymbol{\theta}_m - \boldsymbol{\theta}^*_m) + \frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}^*_m)^{\mathsf{T}} \underbrace{\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}^*|m)}_{=-A}(\boldsymbol{\theta}_m - \boldsymbol{\theta}^*_m)}\ d\boldsymbol{\theta}_m$$

$A = -\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}^*_m|m)$ is the negative Hessian of $\log P(\mathcal{D}, \boldsymbol{\theta}|m)$ evaluated at $\boldsymbol{\theta}^*_m$.

Laplace approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\ d\boldsymbol{\theta}_m$.

As data size N grows (relative to parameter count d), $\boldsymbol{\theta}_m$ becomes more constrained
$\Rightarrow P(\mathcal{D}, \boldsymbol{\theta}_m|m) \propto P(\boldsymbol{\theta}_m|\mathcal{D}, m)$ becomes concentrated on posterior mode $\boldsymbol{\theta}_m^*$.

Idea: approximate log $P(\mathcal{D}, \boldsymbol{\theta}_m|m)$ to second-order around $\boldsymbol{\theta}^*$.

$$
\begin{aligned}
\int P(\mathcal{D}, \boldsymbol{\theta}_m|m) d\boldsymbol{\theta}_m &= \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m|m)}\ d\boldsymbol{\theta}_m \\
&\approx \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m) + \underbrace{\nabla \log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m)}_{=0} \cdot (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*) + \frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)^\mathsf{T} \underbrace{\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}^*|m)}_{=-A} (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)}\ d\boldsymbol{\theta}_m \\
&= \int P(\mathcal{D}, \boldsymbol{\theta}_m^*|m) e^{-\frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)^\mathsf{T} A (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)}\ d\boldsymbol{\theta}_m
\end{aligned}
$$

$A = -\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m)$ is the negative Hessian of log $P(\mathcal{D}, \boldsymbol{\theta}|m)$ evaluated at $\boldsymbol{\theta}_m^*$.

Laplace approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\ d\boldsymbol{\theta}_m$.

As data size N grows (relative to parameter count d), $\boldsymbol{\theta}_m$ becomes more constrained
$\Rightarrow P(\mathcal{D}, \boldsymbol{\theta}_m|m) \propto P(\boldsymbol{\theta}_m|\mathcal{D}, m)$ becomes concentrated on posterior mode $\boldsymbol{\theta}_m^*$.

Idea: approximate log $P(\mathcal{D}, \boldsymbol{\theta}_m|m)$ to second-order around $\boldsymbol{\theta}^*$.

$$
\begin{aligned}
\int P(\mathcal{D}, \boldsymbol{\theta}_m|m) d\boldsymbol{\theta}_m &= \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m|m)}\ d\boldsymbol{\theta}_m \\
&\approx \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m) + \underbrace{\nabla \log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m)}_{=0} \cdot (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*) + \frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)^{\mathsf{T}} \underbrace{\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}^*|m)}_{=-A}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)}\ d\boldsymbol{\theta}_m \\
&= \int P(\mathcal{D}, \boldsymbol{\theta}_m^*|m) e^{-\frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)^{\mathsf{T}} A (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)}\ d\boldsymbol{\theta}_m \\
&= P(\mathcal{D}|\boldsymbol{\theta}_m^*, m) P(\boldsymbol{\theta}_m^*|m) (2\pi)^{\frac{d}{2}} |A|^{-\frac{1}{2}}
\end{aligned}
$$

$A = -\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m)$ is the negative Hessian of log $P(\mathcal{D}, \boldsymbol{\theta}|m)$ evaluated at $\boldsymbol{\theta}_m^*$.

Laplace approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\ d\boldsymbol{\theta}_m$.

As data size N grows (relative to parameter count d), $\boldsymbol{\theta}_m$ becomes more constrained
$\Rightarrow P(\mathcal{D}, \boldsymbol{\theta}_m|m) \propto P(\boldsymbol{\theta}_m|\mathcal{D}, m)$ becomes concentrated on posterior mode $\boldsymbol{\theta}_m^*$.

Idea: approximate log $P(\mathcal{D}, \boldsymbol{\theta}_m|m)$ to second-order around $\boldsymbol{\theta}^*$.

$$\begin{aligned}
\int P(\mathcal{D}, \boldsymbol{\theta}_m|m) d\boldsymbol{\theta}_m &= \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m|m)}\ d\boldsymbol{\theta}_m \\
&\approx \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m) + \underbrace{\nabla \log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m)}_{=0} \cdot (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*) + \frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)^\mathsf{T} \underbrace{\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}^*|m)}_{=-A} (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)}\ d\boldsymbol{\theta}_m \\
&= \int P(\mathcal{D}, \boldsymbol{\theta}_m^*|m) e^{-\frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)^\mathsf{T} A (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)}\ d\boldsymbol{\theta}_m \\
&= P(\mathcal{D}|\boldsymbol{\theta}_m^*, m) P(\boldsymbol{\theta}_m^*|m) (2\pi)^{\frac{d}{2}}\ |A|^{-\frac{1}{2}}
\end{aligned}$$

$A = -\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m)$ is the negative Hessian of $\log P(\mathcal{D}, \boldsymbol{\theta}|m)$ evaluated at $\boldsymbol{\theta}_m^*$.

This is equivalent to approximating the posterior by a Gaussian: an approximation which is asymptotically correct.

Bayesian Information Criterion (BIC)

BIC can be obtained from the Laplace approximation:

$$\log P(\mathcal{D}|m) \approx \log P(\boldsymbol{\theta}^*_m|m) + \log P(\mathcal{D}|\boldsymbol{\theta}^*_m, m) + \frac{d}{2}\log 2\pi - \frac{1}{2}\log|A|$$

We have

$$A = -\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}^*|m) = -\nabla^2 \log P(\mathcal{D}|\boldsymbol{\theta}^*, m) - \nabla^2 \log P(\boldsymbol{\theta}^*|m)$$

So as $N = |\mathcal{D}| \to \infty$, $A \to NA_0 +$ constant, for fixed PD matrix $A_0 = \left\langle -\nabla^2 \log P(\mathbf{x}|\theta^*, m)\right\rangle$.
$\Rightarrow \log|A| \to \log|NA_0| = \log(N^d|A_0|) = d\log N + \log|A_0|$.

Retaining only terms that grow with N we get:

$$\log P(\mathcal{D}|m) \approx \log P(\mathcal{D}|\boldsymbol{\theta}^*_m, m) - \frac{d}{2}\log N$$

Properties:

- Quick and easy to compute.
- Does not depend on prior.
- We can use the ML estimate of θ instead of the MAP estimate (= as $N \to \infty$).
- Related to the "Minimum Description Length" (MDL) criterion (Asst 2).
- Assumes that in the large sample limit, all the parameters are well-determined (i.e. the model is identifiable; otherwise, d should be the number of well-determined parameters).
- Neglects multiple modes (e.g. permutations in a MoG).

Hyperparameters and Evidence optimisation

In some cases, we need to choose between a family of continuously parameterised models.

$$P(\mathcal{D}|\eta) = \int P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\underset{\substack{\uparrow\\ \text{hyperparameters}}}{\eta})\; d\boldsymbol{\theta}$$

This choice can be made by ascending the gradient in:

- the exact evidence (if tractable).
- the approximated evidence (Laplace, EP, Bethe, ...)
- a free-energy bound on the evidence (Variational Bayes)

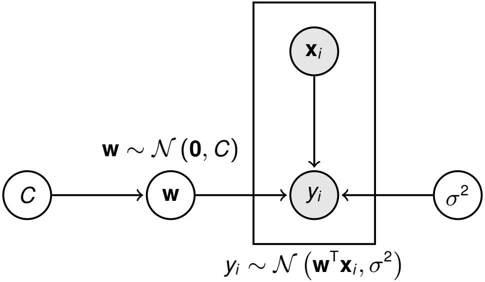
or by placing a hyperprior on the hyperparameters η, and sampling from the posterior

$$P(\eta|\mathcal{D}) = \frac{P(\mathcal{D}|\eta)P(\eta)}{P(\mathcal{D})}$$

using Markov chain Monte Carlo sampling.

Evidence optimisation in linear regression

Consider simple linear regression:



- Maximize

$$P(y_1 \ldots y_N | \mathbf{x}_1 \ldots \mathbf{x}_N, C, \sigma^2) = \int P(y_1 \ldots y_N | \mathbf{x}_1 \ldots \mathbf{x}_N, \mathbf{w}, \sigma^2) P(\mathbf{w}|C) \, d\mathbf{w}$$

  to find optimal values of C, σ.
- Compute the posterior $P(\mathbf{w}|y_1 \ldots y_N, \mathbf{x}_1 \ldots \mathbf{x}_N, C, \sigma^2)$ given these optimal values.

The evidence for linear regression

- The posterior on $\mathbf{w}$ is normal: $\Sigma_{\mathbf{w}} = (\frac{XX^{\mathsf{T}}}{\sigma^2} + C^{-1})^{-1}$; $\bar{\mathbf{w}} = \Sigma_{\mathbf{w}} \frac{XY^{\mathsf{T}}}{\sigma^2}$.

  Note: X is a matrix where columns are input vectors, and Y is a row vector of corresponding predicted outputs.

The evidence for linear regression

- The posterior on $\mathbf{w}$ is normal: $\Sigma_{\mathbf{w}} = (\frac{XX^{\mathsf{T}}}{\sigma^2} + C^{-1})^{-1}$; $\bar{\mathbf{w}} = \Sigma_{\mathbf{w}} \frac{XY^{\mathsf{T}}}{\sigma^2}$.

  Note: X is a matrix where columns are input vectors, and Y is a row vector of corresponding predicted outputs.

- The evidence, $\mathcal{E}(C, \sigma^2) = \int P(Y|X, \mathbf{w}, \sigma^2) P(\mathbf{w}|C)\, d\mathbf{w}$, is given by:

$$\mathcal{E}(C, \sigma^2) = \sqrt{\frac{|2\pi\Sigma_{\mathbf{w}}|}{|2\pi\sigma^2 I|\,|2\pi C|}} \exp\left(-\frac{1}{2} Y \left(\frac{I}{\sigma^2} - \frac{X^{\mathsf{T}}\Sigma_{\mathbf{w}} X}{\sigma^4}\right) Y^{\mathsf{T}}\right)$$

The evidence for linear regression

- The posterior on $\mathbf{w}$ is normal: $\Sigma_{\mathbf{w}} = (\frac{XX^\mathsf{T}}{\sigma^2} + C^{-1})^{-1}$; $\bar{\mathbf{w}} = \Sigma_{\mathbf{w}} \frac{XY^\mathsf{T}}{\sigma^2}$.

  Note: X is a matrix where columns are input vectors, and Y is a row vector of corresponding predicted outputs.

- The evidence, $\mathcal{E}(C, \sigma^2) = \int P(Y|X, \mathbf{w}, \sigma^2) P(\mathbf{w}|C)\, d\mathbf{w}$, is given by:

$$\mathcal{E}(C, \sigma^2) = \sqrt{\frac{|2\pi\Sigma_{\mathbf{w}}|}{|2\pi\sigma^2 I|\;|2\pi C|}} \exp\left(-\frac{1}{2} Y \left(\frac{I}{\sigma^2} - \frac{X^\mathsf{T}\Sigma_{\mathbf{w}}X}{\sigma^4}\right) Y^\mathsf{T}\right)$$

- For optimization, general forms for the gradients are available. If θ is a parameter in C:

$$\frac{\partial}{\partial\theta} \log \mathcal{E}(C, \sigma^2) = \frac{1}{2} \mathrm{Tr}\left[(C - \Sigma_{\mathbf{w}} - \bar{\mathbf{w}}\bar{\mathbf{w}}^\mathsf{T}) \frac{\partial}{\partial\theta} C^{-1}\right]$$

$$\frac{\partial}{\partial\sigma^2} \log \mathcal{E}(C, \sigma^2) = \frac{1}{\sigma^2} \left(-N + \mathrm{Tr}\left[I - \Sigma_{\mathbf{w}} C^{-1}\right] + \frac{1}{\sigma^2} (Y - \bar{\mathbf{w}}^\mathsf{T} X)(Y - \bar{\mathbf{w}}^\mathsf{T} X)^\mathsf{T}\right)$$

Automatic Relevance Determination

The most common form of evidence optimization for regression (due to MacKay and Neal) takes $C^{-1} = \mathrm{diag}(\boldsymbol{\alpha})$ (i.e. $w_i \sim \mathcal{N}(0, \alpha_i^{-1})$) and then optimizes the precisions $\{\alpha_i\}$.

Setting the gradients to 0 and solving gives

$$\alpha_i^{\text{new}} = \frac{1 - \alpha_i [\Sigma_{\mathbf{w}}]_{ii}}{\bar{\mathbf{w}}_i^2}$$

$$(\sigma^2)^{\text{new}} = \frac{(Y - \bar{\mathbf{w}}^{\mathsf{T}} X)(Y - \bar{\mathbf{w}}^{\mathsf{T}} X)^{\mathsf{T}}}{N - \sum_i (1 - [\Sigma_{\mathbf{w}}]_{ii} \alpha_i)}$$

During optimization the α_is meet one of two fates

$$\begin{array}{lcl} \alpha_i \to \infty & \Rightarrow \quad w_i = 0 & \text{irrelevant input } x_i \\ \alpha_i \text{ finite} & \Rightarrow w_i = \operatorname{argmax} \mathrm{P}\,(w_i \mid X, Y, \alpha_i) & \text{relevant input } x_i \end{array}$$

This procedure, Automatic Relevance Determination (ARD), yields sparse solutions that improve on ML regression. (cf. L_1-regression or LASSO).

Evidence optimisation is also called maximum marginal likelihood or ML-2 (Type 2 maximum likelihood).

Prediction averaging

Sometimes, our goal is not to learn the model structure or parameters (or their posteriors); but rather to predict the (conditional) density at a new data point.

Prediction averaging

Sometimes, our goal is not to learn the model structure or parameters (or their posteriors); but rather to predict the (conditional) density at a new data point.

The Bayesian approach in this case should integrate out the parameters:

- Density (unsupervised learning): $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\}$

$$p(\mathbf{x}|\mathcal{D}, m) = \int d\theta \, p(\mathbf{x}|\theta, m) p(\theta|\mathcal{D}, m)$$

- Predictions (supervised learning): $\mathcal{D} = \{(\mathbf{x}_1, y_1)(\mathbf{x}_2, y_2) \ldots (\mathbf{x}_n, y_n)\}$

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}, m) = \int d\theta \, p(\mathbf{y}|\mathbf{x}, \theta, m) p(\theta|\mathcal{D}, m)$$

Prediction averaging

Sometimes, our goal is not to learn the model structure or parameters (or their posteriors); but rather to predict the (conditional) density at a new data point.

The Bayesian approach in this case should integrate out the parameters:

- Density (unsupervised learning): $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\}$

$$p(\mathbf{x}|\mathcal{D}, m) = \int d\theta \, p(\mathbf{x}|\theta, m) p(\theta|\mathcal{D}, m)$$

- Predictions (supervised learning): $\mathcal{D} = \{(\mathbf{x}_1, y_1)(\mathbf{x}_2, y_2) \ldots (\mathbf{x}_n, y_n)\}$

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}, m) = \int d\theta \, p(\mathbf{y}|\mathbf{x}, \theta, m) p(\theta|\mathcal{D}, m)$$

The integral naturally favours predictions associated with large volumes in the posterior, and so incorporates an Occam-like factor.

Prediction averaging

Sometimes, our goal is not to learn the model structure or parameters (or their posteriors); but rather to predict the (conditional) density at a new data point.

The Bayesian approach in this case should integrate out the parameters:

- Density (unsupervised learning): $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\}$

$$p(\mathbf{x}|\mathcal{D}, m) = \int d\theta \, p(\mathbf{x}|\theta, m) p(\theta|\mathcal{D}, m)$$

- Predictions (supervised learning): $\mathcal{D} = \{(\mathbf{x}_1, y_1)(\mathbf{x}_2, y_2) \ldots (\mathbf{x}_n, y_n)\}$

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}, m) = \int d\theta \, p(\mathbf{y}|\mathbf{x}, \theta, m) p(\theta|\mathcal{D}, m)$$

The integral naturally favours predictions associated with large volumes in the posterior, and so incorporates an Occam-like factor.

In principle, predictions may resist overfitting even with an infinitely complex model [or, put another way, the marginalised model has no parameters] $\Rightarrow$ Bayesian nonparametrics.

Prediction averaging

Sometimes, our goal is not to learn the model structure or parameters (or their posteriors); but rather to predict the (conditional) density at a new data point.

The Bayesian approach in this case should integrate out the parameters:

- Density (unsupervised learning): $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_n\}$

$$p(\mathbf{x}|\mathcal{D}, m) = \int d\theta \, p(\mathbf{x}|\theta, m) p(\theta|\mathcal{D}, m)$$

- Predictions (supervised learning): $\mathcal{D} = \{(\mathbf{x}_1, y_1)(\mathbf{x}_2, y_2) \dots (\mathbf{x}_n, y_n)\}$

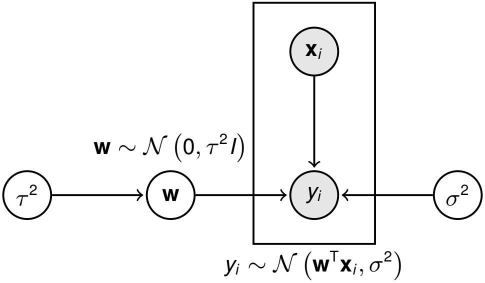
$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}, m) = \int d\theta \, p(\mathbf{y}|\mathbf{x}, \theta, m) p(\theta|\mathcal{D}, m)$$

The integral naturally favours predictions associated with large volumes in the posterior, and so incorporates an Occam-like factor.

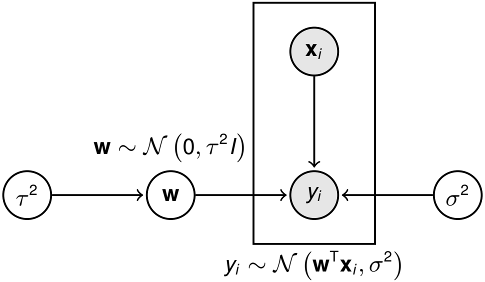
In principle, predictions may resist overfitting even with an infinitely complex model [or, put another way, the marginalised model has no parameters] $\Rightarrow$ Bayesian nonparametrics.

Taking this approach to (non)linear regression leads to a powerful supervised learning method called Gaussian process regression.

Prediction averaging in linear regression



Prediction averaging in linear regression

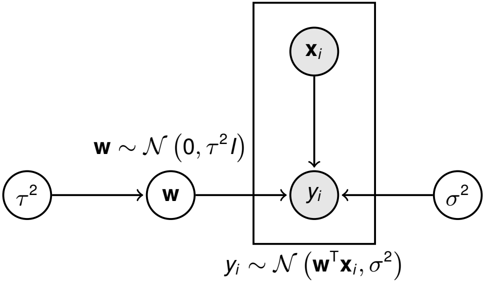


Let $X = [\mathbf{x}_1 \dots \mathbf{x}_N]$, $Y = [y_1 \dots y_N]$. Then (as we've seen)

$$\mathbf{w}|\mathcal{D} \sim \mathcal{N}\left(\bar{\mathbf{w}}, \Sigma_\mathbf{w}\right)$$

where $\Sigma_\mathbf{w} = \left(\frac{1}{\sigma^2} XX^\mathsf{T} + \frac{1}{\tau^2} I\right)^{-1}$ and $\bar{\mathbf{w}} = \frac{1}{\sigma^2} \Sigma_\mathbf{w} X Y^\mathsf{T}$

Prediction averaging in linear regression



Let $X = [\mathbf{x}_1 \ldots \mathbf{x}_N]$, $Y = [y_1 \ldots y_N]$. Then (as we've seen)

$$\mathbf{w}|\mathcal{D} \sim \mathcal{N}\left(\bar{\mathbf{w}}, \Sigma_\mathbf{w}\right)$$

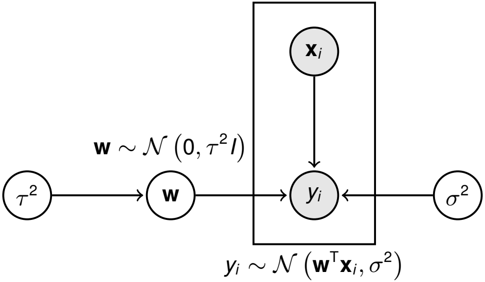
where $\Sigma_\mathbf{w} = \left(\frac{1}{\sigma^2} XX^\mathsf{T} + \frac{1}{\tau^2} I\right)^{-1}$ and $\bar{\mathbf{w}} = \frac{1}{\sigma^2}\Sigma_\mathbf{w} XY^\mathsf{T}$

Thus, given a new input vector $\mathbf{x}$, the predicted output y (integrating out $\mathbf{w}$) is:

$$y|\mathbf{x} \sim \mathcal{N}\left(\bar{\mathbf{w}}^\mathsf{T}\mathbf{x}, \mathbf{x}^\mathsf{T}\Sigma_\mathbf{w}\mathbf{x} + \sigma^2\right).$$

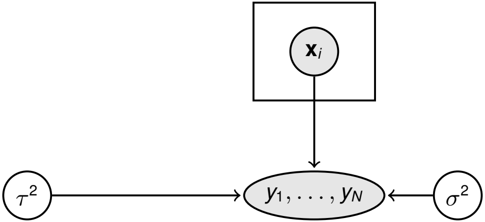
Added variance $\mathbf{x}^\mathsf{T}\Sigma_\mathbf{w}\mathbf{x}$ comes from posterior uncertainty in $\mathbf{w}$ (cf Factor Analysis).

Marginalised linear regression



Integrate out **w** in the model

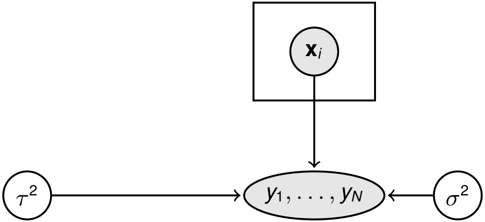
Marginalised linear regression



$$Y^{\mathsf{T}} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

Integrate out $\mathbf{w}$ in the model: the joint distribution of $y_1, \ldots, y_N$ given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is Gaussian.

Marginalised linear regression

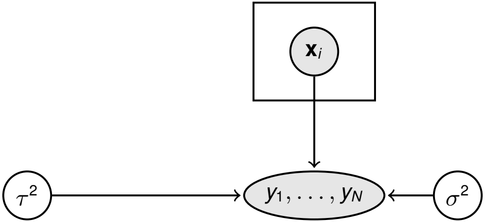


$$Y^{\mathsf{T}} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

Integrate out $\mathbf{w}$ in the model: the joint distribution of $y_1, \ldots, y_N$ given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is Gaussian.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \Bigg| \, \mathbf{x}_1, \ldots, \mathbf{x}_N \sim \mathcal{N}\left(\begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}, \begin{bmatrix} & & & & & & & & & & \\ \\ \\ \\ \end{bmatrix} \right)$$

Marginalised linear regression



$$Y^{\mathsf{T}} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

Integrate out $\mathbf{w}$ in the model: the joint distribution of $y_1, \ldots, y_N$ given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is Gaussian. The means and covariances are:

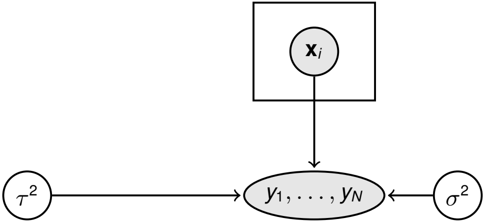
$$E[y_i] =$$

$$E[(y_i - \overline{y_i})^2] =$$

$$E[(y_i - \overline{y_i})(y_j - \overline{y_j})] =$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \Bigg| \mathbf{x}_1, \ldots, \mathbf{x}_N \sim \mathcal{N}\left(\begin{bmatrix} \quad \end{bmatrix}, \begin{bmatrix} \qquad \end{bmatrix} \right)$$

Marginalised linear regression



$$Y^{\mathsf{T}} \sim \mathcal{N}\left(\boldsymbol{\mu}, \Sigma\right)$$

Integrate out $\mathbf{w}$ in the model: the joint distribution of $y_1, \ldots, y_N$ given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is Gaussian. The means and covariances are:

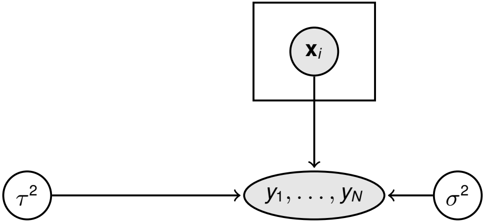
$$E[y_i] = E[\mathbf{w}^{\mathsf{T}}\mathbf{x}_i] = 0^{\mathsf{T}}\mathbf{x}_i = 0$$

$$E[(y_i - \overline{y_i})^2] =$$

$$E[(y_i - \overline{y_i})(y_j - \overline{y_j})] =$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \Bigg| \mathbf{x}_1, \ldots, \mathbf{x}_N \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \quad & \quad & \quad \end{bmatrix} \right)$$

Marginalised linear regression



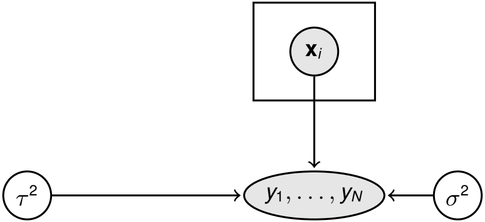
$$Y^\top \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

Integrate out $\mathbf{w}$ in the model: the joint distribution of $y_1, \ldots, y_N$ given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is Gaussian. The means and covariances are:

$$\begin{aligned}
E[y_i] &= E[\mathbf{w}^\top \mathbf{x}_i] = 0^\top \mathbf{x}_i = 0 \\
E[(y_i - \overline{y_i})^2] &= E[(\mathbf{x}_i^\top \mathbf{w})(\mathbf{w}^\top \mathbf{x}_i)] + \sigma^2 = \tau^2 \mathbf{x}_i^\top \mathbf{x}_i + \sigma^2 \\
E[(y_i - \overline{y_i})(y_j - \overline{y_j})] &=
\end{aligned}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \Bigg| \, \mathbf{x}_1, \ldots, \mathbf{x}_N \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 \mathbf{x}_1^\top \mathbf{x}_1 + \sigma^2 & & & \\ & \tau^2 \mathbf{x}_2^\top \mathbf{x}_2 + \sigma^2 & & \\ & & \ddots & \\ & & & \tau^2 \mathbf{x}_N^\top \mathbf{x}_N + \sigma^2 \end{bmatrix} \right)$$

Marginalised linear regression



$$Y^\top \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

Integrate out $\mathbf{w}$ in the model: the joint distribution of $y_1, \ldots, y_N$ given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is Gaussian. The means and covariances are:

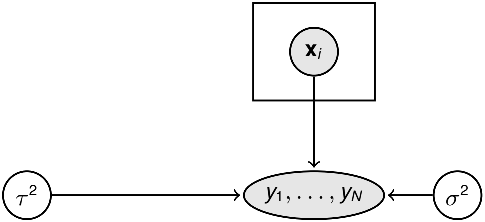
$$E[y_i] = E[\mathbf{w}^\top \mathbf{x}_i] = 0^\top \mathbf{x}_i = 0$$

$$E[(y_i - \overline{y_i})^2] = E[(\mathbf{x}_i^\top \mathbf{w})(\mathbf{w}^\top \mathbf{x}_i)] + \sigma^2 = \tau^2 \mathbf{x}_i^\top \mathbf{x}_i + \sigma^2$$

$$E[(y_i - \overline{y_i})(y_j - \overline{y_j})] = E[(\mathbf{x}_i^\top \mathbf{w})(\mathbf{w}^\top \mathbf{x}_j)] = \tau^2 \mathbf{x}_i^\top \mathbf{x}_j$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \Bigg| \mathbf{x}_1, \ldots, \mathbf{x}_N \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 \mathbf{x}_1^\top \mathbf{x}_1 + \sigma^2 & \tau^2 \mathbf{x}_1^\top \mathbf{x}_2 & \cdots & \tau^2 \mathbf{x}_1^\top \mathbf{x}_N \\ \tau^2 \mathbf{x}_2^\top \mathbf{x}_1 & \tau^2 \mathbf{x}_2^\top \mathbf{x}_2 + \sigma^2 & & \tau^2 \mathbf{x}_2^\top \mathbf{x}_N \\ \vdots & & \ddots & \vdots \\ \tau^2 \mathbf{x}_N^\top \mathbf{x}_1 & \tau^2 \mathbf{x}_N^\top \mathbf{x}_2 & \cdots & \tau^2 \mathbf{x}_N^\top \mathbf{x}_N + \sigma^2 \end{bmatrix} \right)$$

Marginalised linear regression



$$Y^\top \sim \mathcal{N}\left(\mathbf{0}, \tau^2 X^\top X + \sigma^2 I\right)$$

Integrate out $\mathbf{w}$ in the model: the joint distribution of $y_1, \ldots, y_N$ given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is Gaussian. The means and covariances are:

$$E[y_i] = E[\mathbf{w}^\top \mathbf{x}_i] = 0^\top \mathbf{x}_i = 0$$

$$E[(y_i - \overline{y_i})^2] = E[(\mathbf{x}_i^\top \mathbf{w})(\mathbf{w}^\top \mathbf{x}_i)] + \sigma^2 = \tau^2 \mathbf{x}_i^\top \mathbf{x}_i + \sigma^2$$

$$E[(y_i - \overline{y_i})(y_j - \overline{y_j})] = E[(\mathbf{x}_i^\top \mathbf{w})(\mathbf{w}^\top \mathbf{x}_j)] = \tau^2 \mathbf{x}_i^\top \mathbf{x}_j$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \Bigg| \, \mathbf{x}_1, \ldots, \mathbf{x}_N \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 \mathbf{x}_1^\top \mathbf{x}_1 + \sigma^2 & \tau^2 \mathbf{x}_1^\top \mathbf{x}_2 & \cdots & \tau^2 \mathbf{x}_1^\top \mathbf{x}_N \\ \tau^2 \mathbf{x}_2^\top \mathbf{x}_1 & \tau^2 \mathbf{x}_2^\top \mathbf{x}_2 + \sigma^2 & & \tau^2 \mathbf{x}_2^\top \mathbf{x}_N \\ \vdots & & \ddots & \vdots \\ \tau^2 \mathbf{x}_N^\top \mathbf{x}_1 & \tau^2 \mathbf{x}_N^\top \mathbf{x}_2 & \cdots & \tau^2 \mathbf{x}_N^\top \mathbf{x}_N + \sigma^2 \end{bmatrix} \right)$$

Predictions with marginalised regression

Now, include the test input vector $\mathbf{x}$ and test output y:

$$\begin{bmatrix} Y^\top \\ y \end{bmatrix} \bigg| X, \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^\top X + \sigma^2 I & \tau^2 X^\top \mathbf{x} \\ \tau^2 \mathbf{x}^\top X & \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 \end{bmatrix} \right)$$

Predictions with marginalised regression

Now, include the test input vector **x** and test output y:

$$\begin{bmatrix} Y^\top \\ y \end{bmatrix} \Bigg| X, \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^\top X + \sigma^2 I & \tau^2 X^\top \mathbf{x} \\ \tau^2 \mathbf{x}^\top X & \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 \end{bmatrix} \right)$$

We can find $y|Y$ by the standard multivariate Gaussian result:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix} \right) \quad \Rightarrow \quad \mathbf{b}|\mathbf{a} \sim \mathcal{N}\left(K_{BA} K_{AA}^{-1} \mathbf{a}, K_{BB} - K_{BA} K_{AA}{}^{-1} K_{AB} \right)$$

Predictions with marginalised regression

Now, include the test input vector $\mathbf{x}$ and test output y:

$$\begin{bmatrix} Y^\mathsf{T} \\ y \end{bmatrix} \Bigg| X, \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^\mathsf{T} X + \sigma^2 I & \tau^2 X^\mathsf{T} \mathbf{x} \\ \tau^2 \mathbf{x}^\mathsf{T} X & \tau^2 \mathbf{x}^\mathsf{T} \mathbf{x} + \sigma^2 \end{bmatrix} \right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \widetilde{K}_{XX} & \widetilde{K}_{X\mathbf{x}} \\ \widetilde{K}_{\mathbf{x}X} & \widetilde{K}_{\mathbf{x}\mathbf{x}} \end{bmatrix} \right)$$

We can find $y|Y$ by the standard multivariate Gaussian result:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix} \right) \quad \Rightarrow \quad \mathbf{b}|\mathbf{a} \sim \mathcal{N}\left(K_{BA} K_{AA}^{-1} \mathbf{a}, K_{BB} - K_{BA} K_{AA}{}^{-1} K_{AB} \right)$$

So

$$y|Y, X, \mathbf{x} \sim \mathcal{N}\left(\widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} Y^\mathsf{T}, \widetilde{K}_{\mathbf{x}\mathbf{x}} - \widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} \widetilde{K}_{X\mathbf{x}} \right)$$

Predictions with marginalised regression

Now, include the test input vector $\mathbf{x}$ and test output y:

$$\begin{bmatrix} Y^\top \\ y \end{bmatrix} \Bigg| X, \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^\top X + \sigma^2 I & \tau^2 X^\top \mathbf{x} \\ \tau^2 \mathbf{x}^\top X & \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 \end{bmatrix} \right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \widetilde{K}_{XX} & \widetilde{K}_{X\mathbf{x}} \\ \widetilde{K}_{\mathbf{x}X} & \widetilde{K}_{\mathbf{xx}} \end{bmatrix} \right)$$

We can find $y|Y$ by the standard multivariate Gaussian result:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix} \right) \quad \Rightarrow \quad \mathbf{b}|\mathbf{a} \sim \mathcal{N}\left(K_{BA} K_{AA}^{-1} \mathbf{a}, K_{BB} - K_{BA} {K_{AA}}^{-1} K_{AB} \right)$$

So

$$\begin{aligned} y|Y, X, \mathbf{x} &\sim \mathcal{N}\left(\widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} Y^\top, \widetilde{K}_{\mathbf{xx}} - \widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} \widetilde{K}_{X\mathbf{x}} \right) \\ &\sim \mathcal{N}\left(\tau^2 \mathbf{x}^\top X (\tau^2 X^\top X + \sigma^2 I)^{-1} Y^\top, \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 - \tau^2 \mathbf{x}^\top X (\tau^2 X^\top X + \sigma^2 I)^{-1} \tau^2 X^\top \mathbf{x} \right) \end{aligned}$$

Predictions with marginalised regression

Now, include the test input vector $\mathbf{x}$ and test output y:

$$\begin{bmatrix} Y^\top \\ y \end{bmatrix} \Bigg| X, \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^\top X + \sigma^2 I & \tau^2 X^\top \mathbf{x} \\ \tau^2 \mathbf{x}^\top X & \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 \end{bmatrix} \right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \widetilde{K}_{XX} & \widetilde{K}_{X\mathbf{x}} \\ \widetilde{K}_{\mathbf{x}X} & \widetilde{K}_{\mathbf{x}\mathbf{x}} \end{bmatrix} \right)$$

We can find $y|Y$ by the standard multivariate Gaussian result:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix} \right) \quad \Rightarrow \quad \mathbf{b}|\mathbf{a} \sim \mathcal{N}\left(K_{BA} K_{AA}^{-1} \mathbf{a}, K_{BB} - K_{BA} K_{AA}{}^{-1} K_{AB} \right)$$

So

$$\begin{aligned} y|Y, X, \mathbf{x} &\sim \mathcal{N}\left(\widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} Y^\top, \widetilde{K}_{\mathbf{x}\mathbf{x}} - \widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} \widetilde{K}_{X\mathbf{x}} \right) \\ &\sim \mathcal{N}\left(\tau^2 \mathbf{x}^\top X (\tau^2 X^\top X + \sigma^2 I)^{-1} Y^\top, \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 - \tau^2 \mathbf{x}^\top X (\tau^2 X^\top X + \sigma^2 I)^{-1} \tau^2 X^\top \mathbf{x} \right) \\ &\sim \mathcal{N}\left(\mathbf{x}^\top \tfrac{1}{\sigma^2} \Sigma X Y^\top, \mathbf{x}^\top \Sigma \mathbf{x} + \sigma^2 \right) \qquad \Sigma = \left(\tfrac{1}{\sigma^2} X X^\top + \tfrac{1}{\tau^2} I \right)^{-1} \qquad \text{[Matrix Inv. Lem.]} \end{aligned}$$

Predictions with marginalised regression

Now, include the test input vector $\mathbf{x}$ and test output y:

$$\begin{bmatrix} Y^\top \\ y \end{bmatrix} \Big| X, \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^\top X + \sigma^2 I & \tau^2 X^\top \mathbf{x} \\ \tau^2 \mathbf{x}^\top X & \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 \end{bmatrix} \right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \widetilde{K}_{XX} & \widetilde{K}_{X\mathbf{x}} \\ \widetilde{K}_{\mathbf{x}X} & \widetilde{K}_{\mathbf{xx}} \end{bmatrix} \right)$$

We can find $y|Y$ by the standard multivariate Gaussian result:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix} \right) \quad \Rightarrow \quad \mathbf{b}|\mathbf{a} \sim \mathcal{N}\left(K_{BA} K_{AA}^{-1} \mathbf{a}, K_{BB} - K_{BA} K_{AA}{}^{-1} K_{AB} \right)$$

So

$$\begin{aligned} y|Y, X, \mathbf{x} &\sim \mathcal{N}\left(\widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} Y^\top, \widetilde{K}_{\mathbf{xx}} - \widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} \widetilde{K}_{X\mathbf{x}} \right) \\ &\sim \mathcal{N}\left(\tau^2 \mathbf{x}^\top X (\tau^2 X^\top X + \sigma^2 I)^{-1} Y^\top, \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 - \tau^2 \mathbf{x}^\top X (\tau^2 X^\top X + \sigma^2 I)^{-1} \tau^2 X^\top \mathbf{x} \right) \\ &\sim \mathcal{N}\left(\mathbf{x}^\top \underbrace{\tfrac{1}{\sigma^2} \Sigma X Y^\top}_{\bar{\mathbf{w}}}, \mathbf{x}^\top \Sigma \mathbf{x} + \sigma^2 \right) \qquad \Sigma = \underbrace{\left(\tfrac{1}{\sigma^2} X X^\top + \tfrac{1}{\tau^2} I \right)^{-1}}_{\Sigma_{\mathbf{w}}} \qquad \text{[Matrix Inv. Lem.]} \end{aligned}$$

- Same answer as obtained by integrating wrt posterior over $\mathbf{w}$.

Predictions with marginalised regression

Now, include the test input vector $\mathbf{x}$ and test output y:

$$\begin{bmatrix} Y^\top \\ y \end{bmatrix} \Bigg| X, \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^\top X + \sigma^2 I & \tau^2 X^\top \mathbf{x} \\ \tau^2 \mathbf{x}^\top X & \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 \end{bmatrix} \right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \widetilde{K}_{XX} & \widetilde{K}_{X\mathbf{x}} \\ \widetilde{K}_{\mathbf{x}X} & \widetilde{K}_{\mathbf{x}\mathbf{x}} \end{bmatrix} \right)$$

We can find $y|Y$ by the standard multivariate Gaussian result:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix} \right) \quad \Rightarrow \quad \mathbf{b}|\mathbf{a} \sim \mathcal{N}\left(K_{BA} K_{AA}^{-1} \mathbf{a}, K_{BB} - K_{BA} K_{AA}{}^{-1} K_{AB} \right)$$

So

$$\begin{aligned} y|Y, X, \mathbf{x} &\sim \mathcal{N}\left(\widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} Y^\top, \widetilde{K}_{\mathbf{x}\mathbf{x}} - \widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} \widetilde{K}_{X\mathbf{x}} \right) \\ &\sim \mathcal{N}\left(\tau^2 \mathbf{x}^\top X (\tau^2 X^\top X + \sigma^2 I)^{-1} Y^\top, \tau^2 \mathbf{x}^\top \mathbf{x} + \sigma^2 - \tau^2 \mathbf{x}^\top X (\tau^2 X^\top X + \sigma^2 I)^{-1} \tau^2 X^\top \mathbf{x} \right) \\ &\sim \mathcal{N}\left(\mathbf{x}^\top \underbrace{\tfrac{1}{\sigma^2} \Sigma X Y^\top}_{\bar{\mathbf{w}}}, \mathbf{x}^\top \Sigma \mathbf{x} + \sigma^2 \right) \qquad \Sigma = \underbrace{\left(\tfrac{1}{\sigma^2} X X^\top + \tfrac{1}{\tau^2} I \right)^{-1}}_{\Sigma_{\mathbf{w}}} \qquad \text{[Matrix Inv. Lem.]} \end{aligned}$$

- Same answer as obtained by integrating wrt posterior over **w**.
- Evidence $P(Y|X)$ is just joint Gaussian probability; reduces to previous expression.

Predictions with marginalised regression

Now, include the test input vector $\mathbf{x}$ and test output y:

$$\begin{bmatrix} Y^{\mathsf{T}} \\ y \end{bmatrix} \Big| X, \mathbf{x} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^{\mathsf{T}} X + \sigma^2 I & \tau^2 X^{\mathsf{T}} \mathbf{x} \\ \tau^2 \mathbf{x}^{\mathsf{T}} X & \tau^2 \mathbf{x}^{\mathsf{T}} \mathbf{x} + \sigma^2 \end{bmatrix} \right) = \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \widetilde{K}_{XX} & \widetilde{K}_{X\mathbf{x}} \\ \widetilde{K}_{\mathbf{x}X} & \widetilde{K}_{\mathbf{x}\mathbf{x}} \end{bmatrix} \right)$$

We can find $y|Y$ by the standard multivariate Gaussian result:

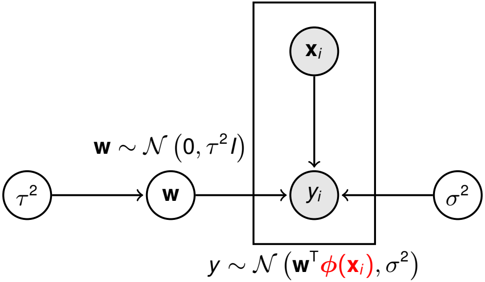
$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix} \right) \quad \Rightarrow \quad \mathbf{b}|\mathbf{a} \sim \mathcal{N}\left(K_{BA} K_{AA}^{-1} \mathbf{a}, K_{BB} - K_{BA} K_{AA}{}^{-1} K_{AB} \right)$$

So

$$\begin{aligned} y|Y, X, \mathbf{x} &\sim \mathcal{N}\left(\widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} Y^{\mathsf{T}}, \widetilde{K}_{\mathbf{x}\mathbf{x}} - \widetilde{K}_{\mathbf{x}X} \widetilde{K}_{XX}^{-1} \widetilde{K}_{X\mathbf{x}} \right) \\ &\sim \mathcal{N}\left(\tau^2 \mathbf{x}^{\mathsf{T}} X (\tau^2 X^{\mathsf{T}} X + \sigma^2 I)^{-1} Y^{\mathsf{T}}, \tau^2 \mathbf{x}^{\mathsf{T}} \mathbf{x} + \sigma^2 - \tau^2 \mathbf{x}^{\mathsf{T}} X (\tau^2 X^{\mathsf{T}} X + \sigma^2 I)^{-1} \tau^2 X^{\mathsf{T}} \mathbf{x} \right) \\ &\sim \mathcal{N}\left(\mathbf{x}^{\mathsf{T}} \underbrace{\tfrac{1}{\sigma^2} \Sigma X Y^{\mathsf{T}}}_{\bar{\mathbf{w}}}, \mathbf{x}^{\mathsf{T}} \Sigma \mathbf{x} + \sigma^2 \right) \qquad \Sigma = \underbrace{\left(\tfrac{1}{\sigma^2} X X^{\mathsf{T}} + \tfrac{1}{\tau^2} I \right)^{-1}}_{\Sigma_{\mathbf{w}}} \qquad \text{[Matrix Inv. Lem.]} \end{aligned}$$

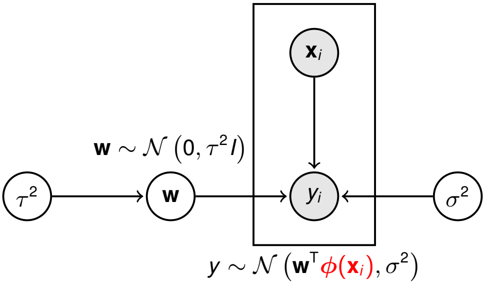
- Same answer as obtained by integrating wrt posterior over **w**.
- Evidence $P(Y|X)$ is just joint Gaussian probability; reduces to previous expression.
- Thus, Bayesian linear regression can be derived from a joint, parameter-free distribution on all the outputs conditioned on all the inputs.

Nonlinear regression



Introduce nonlinear mapping $\mathbf{x} \mapsto \phi(\mathbf{x})$. Each element of $\phi(\mathbf{x})$ is a (nonlinear) feature extracted from $\mathbf{x}$. May be many more features than elements in $\mathbf{x}$.

Nonlinear regression



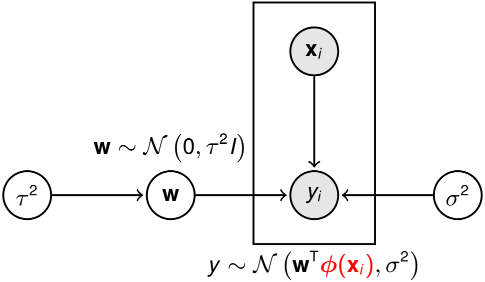
Introduce nonlinear mapping $\mathbf{x} \mapsto \phi(\mathbf{x})$. Each element of $\phi(\mathbf{x})$ is a (nonlinear) feature extracted from $\mathbf{x}$. May be many more features than elements in $\mathbf{x}$.

The regression function $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x})$ is nonlinear, but outputs Y are still jointly Gaussian:

$$Y^\top | X \sim \mathcal{N}(0_N, \tau^2 \Phi^\top \Phi + \sigma^2 I_N)$$

where the i^{th} column of matrix Φ is $\phi(\mathbf{x}_i)$.

Nonlinear regression



Introduce nonlinear mapping $\mathbf{x} \mapsto \phi(\mathbf{x})$. Each element of $\phi(\mathbf{x})$ is a (nonlinear) feature extracted from $\mathbf{x}$. May be many more features than elements in $\mathbf{x}$.

The regression function $f(\mathbf{x}) = \mathbf{w}^\mathsf{T}\phi(\mathbf{x})$ is nonlinear, but outputs Y are still jointly Gaussian:

$$Y^\mathsf{T}|X \sim \mathcal{N}(0_N, \tau^2\Phi^\mathsf{T}\Phi + \sigma^2 I_N)$$

where the i^{th} column of matrix Φ is $\phi(\mathbf{x}_i)$.
Proceeding as before, the predictive distribution over y for a test input $\mathbf{x}$ is:

$$y|\mathbf{x}, Y, X \sim \mathcal{N}\left(\widetilde{K}_{\mathbf{x}X}\widetilde{K}_{XX}^{-1}Y^\mathsf{T}, \widetilde{K}_{\mathbf{xx}} - \widetilde{K}_{\mathbf{x}X}\widetilde{K}_{XX}^{-1}\widetilde{K}_{X\mathbf{x}}\right)$$

where, now $\widetilde{K}_{XX} = \tau^2\Phi^\mathsf{T}\Phi + \sigma^2 I$; $\widetilde{K}_{X\mathbf{x}} = \tau^2\Phi^\mathsf{T}\phi(\mathbf{x})$ and $\widetilde{K}_{\mathbf{xx}} = \tau^2\phi(\mathbf{x})^\mathsf{T}\phi(\mathbf{x}) + \sigma^2$.

The covariance kernel

$$Y^{\mathsf{T}}|X \sim \mathcal{N}\left(\mathbf{0}_N, \tau^2\Phi^{\mathsf{T}}\Phi + \sigma^2 I_N\right)$$

The covariance of the output vector Y plays a central role in the development of the theory of Gaussian processes.

The covariance kernel

$$Y^{\mathsf{T}} | X \sim \mathcal{N}\left(\mathbf{0}_N, \tau^2 \Phi^{\mathsf{T}} \Phi + \sigma^2 I_N\right)$$

The covariance of the output vector Y plays a central role in the development of the theory of Gaussian processes.

Define a covariance kernel function $\widetilde{K} : \mathbb{X} \times \mathbb{X} \mapsto \mathbb{R}$ such that if $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$ are two input vectors with corresponding outputs y, y', then

$$\widetilde{K}(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}[y, y'] = E[yy'] - E[y]E[y']$$

In the nonlinear regression example we have $\widetilde{K}(\mathbf{x}, \mathbf{x}') = \tau^2 \phi(\mathbf{x})^{\mathsf{T}} \phi(\mathbf{x}') + \sigma^2 \delta_{\mathbf{x}=\mathbf{x}'}$.

The covariance kernel

$$Y^{\mathsf{T}}|X \sim \mathcal{N}\left(\mathbf{0}_N, \tau^2 \Phi^{\mathsf{T}} \Phi + \sigma^2 I_N\right)$$

The covariance of the output vector Y plays a central role in the development of the theory of Gaussian processes.

Define a covariance kernel function $\widetilde{K} : \mathbb{X} \times \mathbb{X} \mapsto \mathbb{R}$ such that if $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$ are two input vectors with corresponding outputs y, y', then

$$\widetilde{K}(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}[y, y'] = E[yy'] - E[y]E[y']$$

In the nonlinear regression example we have $\widetilde{K}(\mathbf{x}, \mathbf{x}') = \tau^2 \phi(\mathbf{x})^{\mathsf{T}} \phi(\mathbf{x}') + \sigma^2 \delta_{\mathbf{x}=\mathbf{x}'}$.
Any covariance kernel K has two properties:

- Symmetric: $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$ for all $\mathbf{x}, \mathbf{x}'$.
- Positive semidefinite: the matrix $[K(\mathbf{x}_i, \mathbf{x}_j)]$ formed by any finite set of input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is positive semidefinite.

The covariance kernel

$$Y^\mathsf{T}|X \sim \mathcal{N}\left(\mathbf{0}_N, \tau^2 \Phi^\mathsf{T}\Phi + \sigma^2 I_N\right)$$

The covariance of the output vector Y plays a central role in the development of the theory of Gaussian processes.

Define a covariance kernel function $\widetilde{K} : \mathbb{X} \times \mathbb{X} \mapsto \mathbb{R}$ such that if $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$ are two input vectors with corresponding outputs y, y', then

$$\widetilde{K}(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}[y, y'] = E[yy'] - E[y]E[y']$$

In the nonlinear regression example we have $\widetilde{K}(\mathbf{x}, \mathbf{x}') = \tau^2 \phi(\mathbf{x})^\mathsf{T} \phi(\mathbf{x}') + \sigma^2 \delta_{\mathbf{x}=\mathbf{x}'}$.
Any covariance kernel K has two properties:

- Symmetric: $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$ for all $\mathbf{x}, \mathbf{x}'$.
- Positive semidefinite: the matrix $[K(\mathbf{x}_i, \mathbf{x}_j)]$ formed by any finite set of input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is positive semidefinite.

Theorem: A covariance kernel $K : \mathbb{X} \times \mathbb{X} \mapsto \mathbb{R}$ is symmetric and positive semidefinite if and only if there is a feature map $\phi : \mathbb{X} \mapsto \mathbb{H}$ such that

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\mathsf{T} \phi(\mathbf{x}')$$

The feature space $\mathbb{H}$ might be an infinite-dimensional Hilbert space.

Regression using the covariance kernel

For Bayesian regression, prediction depends on $\widetilde{K}(\mathbf{x}, \mathbf{x})$ rather than explicitly on $\phi(\mathbf{x})$.

Regression using the covariance kernel

For Bayesian regression, prediction depends on $\widetilde{K}(\mathbf{x}, \mathbf{x})$ rather than explicitly on $\phi(\mathbf{x})$.

So we can define the joint in terms of $\widetilde{K}$ *implicitly* using a (potentially infinite-dimensional) feature map $\phi(\mathbf{x})$.

$$Y|X, \widetilde{K} \sim \mathcal{N}(0_N, \widetilde{K}_{XX}))$$

where the i, j entry in the covariance matrix $\widetilde{K}_{XX}$ is $\widetilde{K}(\mathbf{x}_i, \mathbf{x}_j)$.

This is the **kernel trick**.

Regression using the covariance kernel

For Bayesian regression, prediction depends on $\widetilde{K}(\mathbf{x}, \mathbf{x})$ rather than explicitly on $\phi(\mathbf{x})$.

So we can define the joint in terms of $\widetilde{K}$ *implicitly* using a (potentially infinite-dimensional) feature map $\phi(\mathbf{x})$.

$$Y|X, \widetilde{K} \sim \mathcal{N}(0_N, \widetilde{K}_{XX}))$$

where the i, j entry in the covariance matrix $\widetilde{K}_{XX}$ is $\widetilde{K}(\mathbf{x}_i, \mathbf{x}_j)$.

This is the **kernel trick**.

Prediction: compute the predictive distribution of y conditioned on Y as before:

$$y|\mathbf{x}, X, Y, \widetilde{K} \sim \mathcal{N}\left(\widetilde{K}_{X\mathbf{x}}^{\mathsf{T}} \widetilde{K}_{XX}^{-1} Y^{\mathsf{T}}, \widetilde{K}_{\mathbf{xx}} - \widetilde{K}_{X\mathbf{x}}^{\mathsf{T}} \widetilde{K}_{XX}^{-1} \widetilde{K}_{X\mathbf{x}}\right)$$

where now $[\widetilde{K}_{XX}]_{ij} = \widetilde{K}(\mathbf{x}_i, \mathbf{x}_j)$; $[\widetilde{K}_{X\mathbf{x}}]_i = \widetilde{K}(\mathbf{x}_i, \mathbf{x})$ and $\widetilde{K}_{\mathbf{xx}} = \widetilde{K}(\mathbf{x}, \mathbf{x})$.

Regression using the covariance kernel

For Bayesian regression, prediction depends on $\widetilde{K}(\mathbf{x}, \mathbf{x})$ rather than explicitly on $\phi(\mathbf{x})$.

So we can define the joint in terms of $\widetilde{K}$ *implicitly* using a (potentially infinite-dimensional) feature map $\phi(\mathbf{x})$.

$$Y|X, \widetilde{K} \sim \mathcal{N}(0_N, \widetilde{K}_{XX}))$$

where the i, j entry in the covariance matrix $\widetilde{K}_{XX}$ is $\widetilde{K}(\mathbf{x}_i, \mathbf{x}_j)$.

This is the **kernel trick**.

Prediction: compute the predictive distribution of y conditioned on Y as before:

$$y|\mathbf{x}, X, Y, \widetilde{K} \sim \mathcal{N}\left(\widetilde{K}_{X\mathbf{x}}^{\mathsf{T}} \widetilde{K}_{XX}^{-1} Y^{\mathsf{T}}, \widetilde{K}_{\mathbf{xx}} - \widetilde{K}_{X\mathbf{x}}^{\mathsf{T}} \widetilde{K}_{XX}^{-1} \widetilde{K}_{X\mathbf{x}}\right)$$

where now $[\widetilde{K}_{XX}]_{ij} = \widetilde{K}(\mathbf{x}_i, \mathbf{x}_j)$; $[\widetilde{K}_{X\mathbf{x}}]_i = \widetilde{K}(\mathbf{x}_i, \mathbf{x})$ and $\widetilde{K}_{\mathbf{xx}} = \widetilde{K}(\mathbf{x}, \mathbf{x})$.

Evidence: given by the Gaussian likelihood:

$$P(Y|X, \widetilde{K}) = |2\pi \widetilde{K}_{XX}|^{-\frac{1}{2}} e^{-\frac{1}{2} Y \widetilde{K}_{XX}^{-1} Y^{\mathsf{T}}}$$

Regression using the covariance kernel

For Bayesian regression, prediction depends on $\widetilde{K}(\mathbf{x}, \mathbf{x})$ rather than explicitly on $\phi(\mathbf{x})$.

So we can define the joint in terms of $\widetilde{K}$ *implicitly* using a (potentially infinite-dimensional) feature map $\phi(\mathbf{x})$.

$$Y|X, \widetilde{K} \sim \mathcal{N}(0_N, \widetilde{K}_{XX}))$$

where the i, j entry in the covariance matrix $\widetilde{K}_{XX}$ is $\widetilde{K}(\mathbf{x}_i, \mathbf{x}_j)$.

This is the **kernel trick**.

Prediction: compute the predictive distribution of y conditioned on Y as before:

$$y|\mathbf{x}, X, Y, \widetilde{K} \sim \mathcal{N}\left(\widetilde{K}_{X\mathbf{x}}^{\mathsf{T}} \widetilde{K}_{XX}^{-1} Y^{\mathsf{T}}, \widetilde{K}_{\mathbf{xx}} - \widetilde{K}_{X\mathbf{x}}^{\mathsf{T}} \widetilde{K}_{XX}^{-1} \widetilde{K}_{X\mathbf{x}}\right)$$

where now $[\widetilde{K}_{XX}]_{ij} = \widetilde{K}(\mathbf{x}_i, \mathbf{x}_j)$; $[\widetilde{K}_{X\mathbf{x}}]_i = \widetilde{K}(\mathbf{x}_i, \mathbf{x})$ and $\widetilde{K}_{\mathbf{xx}} = \widetilde{K}(\mathbf{x}, \mathbf{x})$.

Evidence: given by the Gaussian likelihood:

$$P(Y|X, \widetilde{K}) = |2\pi \widetilde{K}_{XX}|^{-\frac{1}{2}} e^{-\frac{1}{2} Y \widetilde{K}_{XX}^{-1} Y^{\mathsf{T}}}$$

Evidence optimisation: the covariance kernel $\widetilde{K}$ often has (hyper)parameters, and these can be optimized by gradient ascent in $\log P(Y|X, \widetilde{K})$.

The Gaussian process

A covariance kernel $K(\mathbf{x}, \mathbf{x}')$ (and mean function $m(\mathbf{x})$) defined on a domain $\mathbb{X}$ defines a **Gaussian process** (GP): a stochastic process (ie collection of random variables) on $\mathbb{R}$ indexed by $\mathbf{x} \in \mathbb{X}$, any finite subset of which have (consistent) Gaussian distributions.

The Gaussian process

A covariance kernel $K(\mathbf{x}, \mathbf{x}')$ (and mean function $m(\mathbf{x})$) defined on a domain $\mathbb{X}$ defines a **Gaussian process** (GP): a stochastic process (ie collection of random variables) on $\mathbb{R}$ indexed by $\mathbf{x} \in \mathbb{X}$, any finite subset of which have (consistent) Gaussian distributions.

Let $f(\mathbf{x})$ be the random variable indexed by $\mathbf{x}$. Then a draw from the whole GP (ie for all $\mathbf{x}$) is a random **function** $f : \mathbb{X} \mapsto \mathbb{R}$. We write

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), K(\cdot, \cdot))$$

where the $(\cdot)$s emphasise that these are all functions.

The Gaussian process

A covariance kernel $K(\mathbf{x}, \mathbf{x}')$ (and mean function $m(\mathbf{x})$) defined on a domain $\mathbb{X}$ defines a **Gaussian process** (GP): a stochastic process (ie collection of random variables) on $\mathbb{R}$ indexed by $\mathbf{x} \in \mathbb{X}$, any finite subset of which have (consistent) Gaussian distributions.

Let $f(\mathbf{x})$ be the random variable indexed by $\mathbf{x}$. Then a draw from the whole GP (ie for all $\mathbf{x}$) is a random **function** $f : \mathbb{X} \mapsto \mathbb{R}$. We write

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), K(\cdot, \cdot))$$

where the $(\cdot)$s emphasise that these are all functions.

The GP is defined such that, given a finite list of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the joint distribution of the function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\mathsf{T}$ is:

$$\mathbf{f}|X, K \sim \mathcal{N}(\mathbf{m}, K_{XX})$$

where, as usual, $[K_{XX}]_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $[\mathbf{m}]_i = m(\mathbf{x}_i)$. If we enlarge or reduce the set of $\mathbf{x}_i$s then the means and covariance matrices produced by fixed functions marginalise correctly.

The Gaussian process

A covariance kernel $K(\mathbf{x}, \mathbf{x}')$ (and mean function $m(\mathbf{x})$) defined on a domain $\mathbb{X}$ defines a **Gaussian process** (GP): a stochastic process (ie collection of random variables) on $\mathbb{R}$ indexed by $\mathbf{x} \in \mathbb{X}$, any finite subset of which have (consistent) Gaussian distributions.

Let $f(\mathbf{x})$ be the random variable indexed by $\mathbf{x}$. Then a draw from the whole GP (ie for all $\mathbf{x}$) is a random **function** $f : \mathbb{X} \mapsto \mathbb{R}$. We write

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), K(\cdot, \cdot))$$

where the $(\cdot)$s emphasise that these are all functions.

The GP is defined such that, given a finite list of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the joint distribution of the function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$ is:

$$\mathbf{f} | X, K \sim \mathcal{N}(\mathbf{m}, K_{XX})$$

where, as usual, $[K_{XX}]_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $[\mathbf{m}]_i = m(\mathbf{x}_i)$. If we enlarge or reduce the set of $\mathbf{x}_i$s then the means and covariance matrices produced by fixed functions marginalise correctly.

For nonlinear regression, $f(\cdot)$ could instead be defined by drawing the weight vector $\mathbf{w} \in \mathbb{H}$ from the prior. But $\mathbb{H}$ may be infinite dimensional $\Rightarrow$ need an infinite-size object to make even a single prediction. In the GP view, each $f(\mathbf{x})$ can be drawn separately.

Regression with Gaussian processes

We seek to learn a function that maps inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$ to outputs $y_1, \ldots, y_N$.

Regression with Gaussian processes

We seek to learn a function that maps inputs $\mathbf{x}_1, \dots, \mathbf{x}_N$ to outputs $y_1, \dots, y_N$.

Instead of assuming a specific form, consider a random function drawn from a GP prior:

$$f(\cdot) \sim \mathcal{GP}(0, K(\cdot, \cdot))\,.$$

Any function is possible (no restriction on support) but some are (much) more likely *a priori*.

Regression with Gaussian processes

We seek to learn a function that maps inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$ to outputs $y_1, \ldots, y_N$.

Instead of assuming a specific form, consider a random function drawn from a GP prior:

$$f(\cdot) \sim \mathcal{GP}(0, K(\cdot, \cdot)) .$$

Any function is possible (no restriction on support) but some are (much) more likely *a priori*.

Observations y_i are taken to be noisy versions of the (almost surely continuous) latent $f(\mathbf{x}_i)$:

$$y_i | \mathbf{x}_i, f(\cdot) \sim \mathcal{N}\left(f(\mathbf{x}_i), \sigma^2\right) \qquad [\text{so } Y \sim \mathcal{N}\left(0, \widetilde{K}_{XX}\right) \text{ with } \widetilde{K}_{XX} = K_{XX} + \sigma^2 I]$$

Regression with Gaussian processes

We seek to learn a function that maps inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$ to outputs $y_1, \ldots, y_N$.

Instead of assuming a specific form, consider a random function drawn from a GP prior:

$$f(\cdot) \sim \mathcal{GP}(0, K(\cdot,\cdot)) \, .$$

Any function is possible (no restriction on support) but some are (much) more likely *a priori*.

Observations y_i are taken to be noisy versions of the (almost surely continuous) latent $f(\mathbf{x}_i)$:

$$y_i|\mathbf{x}_i, f(\cdot) \sim \mathcal{N}\left(f(\mathbf{x}_i), \sigma^2\right) \qquad [\text{so } Y \sim \mathcal{N}\left(0, \widetilde{K}_{XX}\right) \text{ with } \widetilde{K}_{XX} = K_{XX} + \sigma^2 I]$$

Evidence: given by the multivariate Gaussian likelihood:

$$P(Y|X) = |2\pi(K_{XX} + \sigma^2 I)|^{-\frac{1}{2}} e^{-\frac{1}{2} Y (K_{XX} + \sigma^2 I)^{-1} Y^{\mathsf{T}}}$$

Regression with Gaussian processes

We seek to learn a function that maps inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$ to outputs $y_1, \ldots, y_N$.

Instead of assuming a specific form, consider a random function drawn from a GP prior:

$$f(\cdot) \sim \mathcal{GP}(0, K(\cdot, \cdot)) .$$

Any function is possible (no restriction on support) but some are (much) more likely *a priori*.

Observations y_i are taken to be noisy versions of the (almost surely continuous) latent $f(\mathbf{x}_i)$:

$$y_i | \mathbf{x}_i, f(\cdot) \sim \mathcal{N}\left(f(\mathbf{x}_i), \sigma^2\right) \qquad [\text{so } Y \sim \mathcal{N}\left(0, \widetilde{K}_{XX}\right) \text{ with } \widetilde{K}_{XX} = K_{XX} + \sigma^2 I]$$

Evidence: given by the multivariate Gaussian likelihood:

$$P(Y|X) = |2\pi(K_{XX} + \sigma^2 I)|^{-\frac{1}{2}} e^{-\frac{1}{2} Y (K_{XX} + \sigma^2 I)^{-1} Y^{\mathsf{T}}}$$

Posterior: on latent f is also a GP:

$$f(\cdot)|X, Y \sim \mathcal{GP}(K_{\cdot X}(K_{XX} + \sigma^2 I)^{-1} Y^{\mathsf{T}}, K(\cdot, \cdot) - K_{\cdot X}(K_{XX} + \sigma^2 I)^{-1} K_{X \cdot}))$$

Regression with Gaussian processes

We seek to learn a function that maps inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$ to outputs $y_1, \ldots, y_N$.

Instead of assuming a specific form, consider a random function drawn from a GP prior:

$$f(\cdot) \sim \mathcal{GP}(0, K(\cdot, \cdot))\,.$$

Any function is possible (no restriction on support) but some are (much) more likely *a priori*.

Observations y_i are taken to be noisy versions of the (almost surely continuous) latent $f(\mathbf{x}_i)$:

$$y_i|\mathbf{x}_i, f(\cdot) \sim \mathcal{N}\left(f(\mathbf{x}_i), \sigma^2\right) \qquad [\text{so } Y \sim \mathcal{N}\left(0, \widetilde{K}_{XX}\right) \text{ with } \widetilde{K}_{XX} = K_{XX} + \sigma^2 I]$$

Evidence: given by the multivariate Gaussian likelihood:

$$P(Y|X) = |2\pi(K_{XX} + \sigma^2 I)|^{-\frac{1}{2}} e^{-\frac{1}{2} Y (K_{XX}+\sigma^2 I)^{-1} Y^{\mathsf{T}}}$$

Posterior: on latent f is also a GP:

$$f(\cdot)|X, Y \sim \mathcal{GP}(K_{\cdot X}(K_{XX} + \sigma^2 I)^{-1} Y^{\mathsf{T}}, K(\cdot, \cdot) - K_{\cdot X}(K_{XX} + \sigma^2 I)^{-1} K_{X\cdot}))$$

Predictions: posterior on f, plus observation noise:

$$y|X, Y, \mathbf{x} \sim \mathcal{N}\left(E[f(\mathbf{x})|X, Y], \mathrm{Var}[f(\mathbf{x})|X, Y] + \sigma^2\right) = \mathcal{N}\left(K_{\mathbf{x}X}\widetilde{K}_{XX}^{-1} Y, K_{\mathbf{x}X}\widetilde{K}_{XX}^{-1} K_{X\mathbf{x}} + \sigma^2\right)$$

Regression with Gaussian processes

We seek to learn a function that maps inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$ to outputs $y_1, \ldots, y_N$.

Instead of assuming a specific form, consider a random function drawn from a GP prior:

$$f(\cdot) \sim \mathcal{GP}(0, K(\cdot, \cdot))\,.$$

Any function is possible (no restriction on support) but some are (much) more likely *a priori*.

Observations y_i are taken to be noisy versions of the (almost surely continuous) latent $f(\mathbf{x}_i)$:

$$y_i|\mathbf{x}_i, f(\cdot) \sim \mathcal{N}\left(f(\mathbf{x}_i), \sigma^2\right) \qquad [\text{so } Y \sim \mathcal{N}\left(0, \widetilde{K}_{XX}\right) \text{ with } \widetilde{K}_{XX} = K_{XX} + \sigma^2 I]$$

Evidence: given by the multivariate Gaussian likelihood:

$$P(Y|X) = |2\pi(K_{XX} + \sigma^2 I)|^{-\frac{1}{2}} e^{-\frac{1}{2} Y(K_{XX}+\sigma^2 I)^{-1} Y^{\mathsf{T}}}$$

Posterior: on latent f is also a GP:

$$f(\cdot)|X, Y \sim \mathcal{GP}(K_{\cdot X}(K_{XX} + \sigma^2 I)^{-1} Y^{\mathsf{T}}, K(\cdot, \cdot) - K_{\cdot X}(K_{XX} + \sigma^2 I)^{-1} K_{X \cdot}))$$

Predictions: posterior on f, plus observation noise:

$$y|X, Y, \mathbf{x} \sim \mathcal{N}\left(E[f(\mathbf{x})|X, Y], \mathrm{Var}[f(\mathbf{x})|X, Y] + \sigma^2\right) = \mathcal{N}\left(K_{\mathbf{x}X}\widetilde{K}_{XX}^{-1} Y, K_{\mathbf{x}X}\widetilde{K}_{XX}^{-1} K_{X\mathbf{x}} + \sigma^2\right)$$

Evidence Optimisation: gradient ascent in $\log P(Y|X)$.

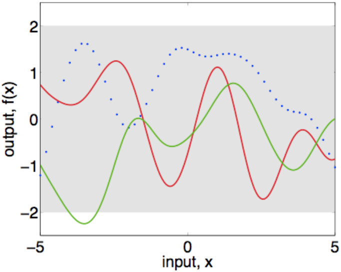
Samples from a Gaussian process

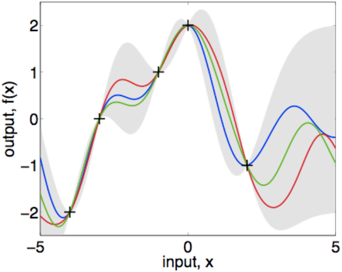
We can draw sample functions from a GP by fixing a set of input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$, and drawing a sample $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)$ from the corresponding multivariate Gaussian.

Samples from a Gaussian process

We can draw sample functions from a GP by fixing a set of input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$, and drawing a sample $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)$ from the corresponding multivariate Gaussian.

Example prior and posterior GPs:

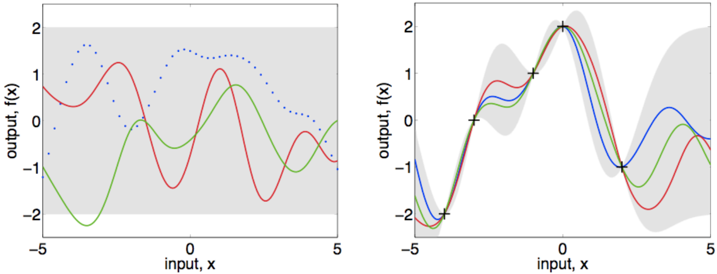




Samples from a Gaussian process

We can draw sample functions from a GP by fixing a set of input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$, and drawing a sample $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)$ from the corresponding multivariate Gaussian.

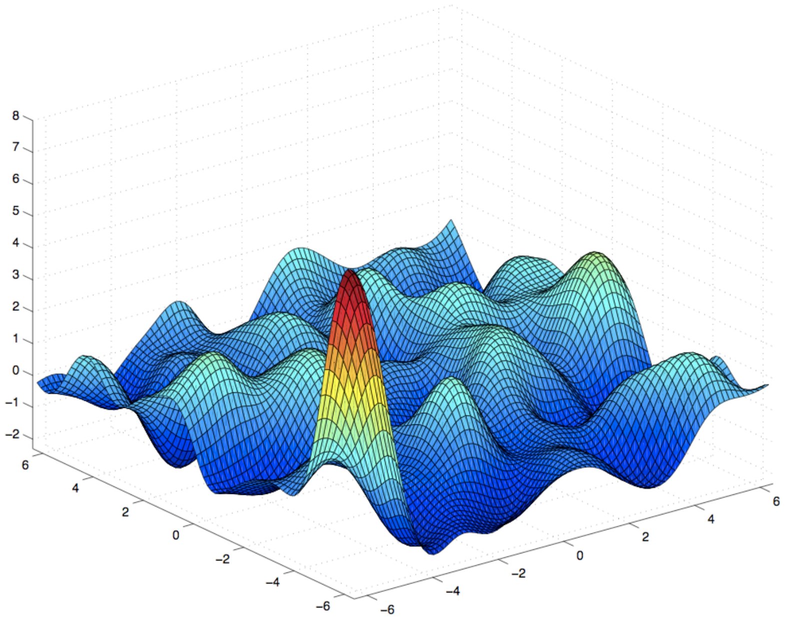
Example prior and posterior GPs:



Another approach is to

- sample $f(\mathbf{x}_1)$ first,
- then $f(\mathbf{x}_2)|f(\mathbf{x}_1)$,
- and generally $f(\mathbf{x}_n)|f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{n-1})$ for $n = 1, 2, \ldots$.

Sample from a 2D Gaussian process

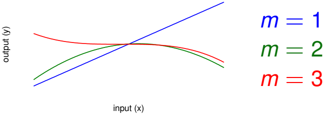


Examples of covariance kernels

- Polynomial:

$$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\mathsf{T}\mathbf{x}')^m \qquad m = 1, 2, \ldots$$

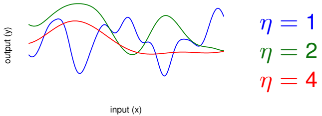
f is inhomogeneous polynomial of degree m



- Squared-exponential (or exponentiated-quadratic):

$$K(\mathbf{x}, \mathbf{x}') = \theta^2 e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\eta^2}}$$

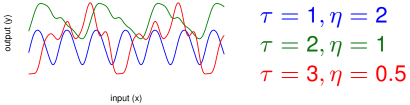
f is smooth (C^∞ almost surely) on length scale η



- Periodic (exp-sine):

$$K(x, x') = \theta^2 e^{-\frac{2\sin^2(\pi(x-x')/\tau)}{\eta^2}}$$

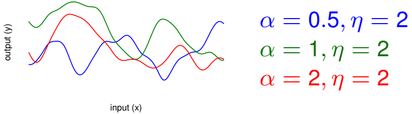
f is smooth and periodic



- Rational Quadratic:

$$K(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\alpha\eta^2}\right)^{-\alpha} \qquad \alpha > 0$$

f is smooth over multiple scales



Forms of kernels

If K_1 and K_2 are covariance kernels, then so are:

- Rescaling: αK_1 for $\alpha > 0$.
- Addition: $K_1 + K_2$
- Elementwise product: $K_1 K_2$
- Mapping: $K_1(\phi(\mathbf{x}), \phi(\mathbf{x}'))$ for some function ϕ.

Forms of kernels

If K_1 and K_2 are covariance kernels, then so are:

- Rescaling: αK_1 for $\alpha > 0$.
- Addition: $K_1 + K_2$
- Elementwise product: $K_1 K_2$
- Mapping: $K_1(\phi(\mathbf{x}), \phi(\mathbf{x}'))$ for some function ϕ.

A covariance kernel is translation-invariant if

$$K(\mathbf{x}, \mathbf{x}') = h(\mathbf{x} - \mathbf{x}')$$

A GP with a translation-invariant covariance kernel is stationary: if $f(\cdot) \sim \mathcal{GP}(0, K)$, then so is $f(\cdot - \mathbf{x}) \sim \mathcal{GP}(0, K)$ for each $\mathbf{x}$.

Forms of kernels

If K_1 and K_2 are covariance kernels, then so are:

- Rescaling: αK_1 for $\alpha > 0$.
- Addition: $K_1 + K_2$
- Elementwise product: $K_1 K_2$
- Mapping: $K_1(\phi(\mathbf{x}), \phi(\mathbf{x}'))$ for some function ϕ.

A covariance kernel is translation-invariant if

$$K(\mathbf{x}, \mathbf{x}') = h(\mathbf{x} - \mathbf{x}')$$

A GP with a translation-invariant covariance kernel is stationary: if $f(\cdot) \sim \mathcal{GP}(0, K)$, then so is $f(\cdot - \mathbf{x}) \sim \mathcal{GP}(0, K)$ for each $\mathbf{x}$.

A covariance kernel is radial or radially symmetric if

$$K(\mathbf{x}, \mathbf{x}') = h(\|\mathbf{x} - \mathbf{x}'\|)$$

A GP with a radial covariance kernel is stationary with respect to translations, rotations, and reflections of the input space.

GP methods

- With suitable kernels, combinations of kernels, and hyperparameter learning, GPs can identify a wide range of functional dependence. (The “automated statistician” project starts with GPs).

- With approximation, the mapping from f to y may be taken to be non-Gaussian, allowing GP classication, ordinal regression, domain-specific noise and more.

- Functions in more complex hierarchical models may be drawn from GP priors:
  - GP latent variable model (GPLVM)
  - Stacked GPs
  - Deep GP networks

- Inference and learning require inversion of K_{XX}: scales as N^3. Sparse approximate methods reduce this to order N.

- State-of-the-art approach, particularly when data are limited.

Nonparametric Bayesian Models and Occam's Razor

Overparameterised models can overfit. In the GP, the "parameter" is the function $f(\mathbf{x})$ (or "weights" in non-linear feature space) which can be infinite-dimensional.

Nonparametric Bayesian Models and Occam's Razor

Overparameterised models can overfit. In the GP, the "parameter" is the function $f(\mathbf{x})$ (or "weights" in non-linear feature space) which can be infinite-dimensional.

However, the Bayesian treatment integrates over these parameters: we never identify a single "best fit" f, just a posterior (and posterior mean). So f cannot be adjusted to overfit the data.

Nonparametric Bayesian Models and Occam's Razor

Overparameterised models can overfit. In the GP, the "parameter" is the function $f(\mathbf{x})$ (or "weights" in non-linear feature space) which can be infinite-dimensional.

However, the Bayesian treatment integrates over these parameters: we never identify a single "best fit" f, just a posterior (and posterior mean). So f cannot be adjusted to overfit the data.

The GP is an example of the larger class of **nonparametric Bayesian models**.

- Infinite number of parameters.
- Often constructed as the infinite limit of a nested family of finite models (sometimes equivalent to infinite model averaging).
- Parameters integrated out, so effective number of parameters to overfit is zero or small (hyperparameters).
- No need for model selection. Bayesian posterior on parameters will concentrate on "submodel" with largest integral automatically.
- No explicit need for Occam's razor, validation or added regularisation penalty.
- Examples include the Dirichlet process (infinite mixtures), Infinite Binary Prior (infinite binary factor models), Infinite HMM . . .